

MMM		MMM	AAAAAAAAAA	CCCCCCCCCCCC	RRRRRRRRRRRR	0000000000		
MMM		MMM	AAAAAAAAAA	CCCCCCCCCCCC	RRRRRRRRRRRR	0000000000		
MMM		MMM	AAAAAAAAAA	CCCCCCCCCCCC	RRRRRRRRRRRR	0000000000		
MMMMMMM	MMMMMMM	AAA	AAA	CCC	RRR	RRR	000	000
MMMMMMM	MMMMMMM	AAA	AAA	CCC	RRR	RRR	000	000
MMMMMMM	MMMMMMM	AAA	AAA	CCC	RRR	RRR	000	000
MMM	MMM	MMM	AAA	AAA	CCC	RRR	RRR	000
MMM	MMM	MMM	AAA	AAA	CCC	RRR	RRR	000
MMM	MMM	MMM	AAA	AAA	CCC	RRR	RRR	000
MMM		MMM	AAA	AAA	CCC	RRRRRRRRRRRR	000	000
MMM		MMM	AAA	AAA	CCC	RRRRRRRRRRRR	000	000
MMM		MMM	AAA	AAA	CCC	RRRRRRRRRRRR	000	000
MMM		MMM	AAAAAAAAAAAAAAAA	CCC	RRR	RRR	000	000
MMM		MMM	AAAAAAAAAAAAAAAA	CCC	RRR	RRR	000	000
MMM		MMM	AAAAAAAAAAAAAAAA	CCC	RRR	RRR	000	000
MMM		MMM	AAA	AAA	CCC	RRR	RRR	000
MMM		MMM	AAA	AAA	CCC	RRR	RRR	000
MMM		MMM	AAA	AAA	CCC	RRR	RRR	000
MMM		MMM	AAA	AAA	CCCCCCCCCCCC	RRR	RRR	0000000000
MMM		MMM	AAA	AAA	CCCCCCCCCCCC	RRR	RRR	0000000000
MMM		MMM	AAA	AAA	CCCCCCCCCCCC	RRR	RRR	0000000000

```

LL          IIIIII          SSSSSSSS
LL          IIIIII          SSSSSSSS
LL          II             SS
LL          II             SS
LL          II             SS
LL          II             SS
LL          II             SSSSSS
LL          II             SSSSSS
LL          II             SS
LL          II             SS
LL          II             SS
LL          II             SS
LLLLLLLLLLLL IIIIII          SSSSSSSS
LLLLLLLLLLLL IIIIII          SSSSSSSS

```


(2)	79	DECLARATIONS
(3)	100	ERRORS FOUND BY THE GRAMMAR/PARSER
(4)	180	IDENT PROCESS .IDENT STATEMENT
(5)	226	TITLE PROCESS .TITLE STATEMENT
(6)	272	SBTTL PROCESS .SBTTL STATEMENT
(7)	370	ENABL/DSABL PROCESS .ENABL/.DSABL
(8)	422	LIST/NLIST PROCESS .LIST/.NLIST
(9)	487	PROCESS .CROSS/.NOCROSS DIRECTIVES
(10)	550	SETDFL PROCESS .DEFAULT DIRECTIVE
(11)	599	ENDPRG PROCESS .END STATEMENT

```
0000 1
0000 2      .TITLE  MAC$ACTONE ONCE-ONLY ACTION ROUTINES
0000 3      .IDENT  'V04-000'
0000 4
0000 5
0000 6      :*****
0000 7      :*
0000 8      :*  COPYRIGHT (c) 1978, 1980, 1982, 1984 BY
0000 9      :*  DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.
0000 10     :*  ALL RIGHTS RESERVED.
0000 11     :*
0000 12     :*  THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED
0000 13     :*  ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE
0000 14     :*  INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR ANY OTHER
0000 15     :*  COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY
0000 16     :*  OTHER PERSON. NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY
0000 17     :*  TRANSFERRED.
0000 18     :*
0000 19     :*  THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE
0000 20     :*  AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT
0000 21     :*  CORPORATION.
0000 22     :*
0000 23     :*  DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS
0000 24     :*  SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.
0000 25     :*
0000 26     :*
0000 27     :*****
0000 28     :
0000 29     :
0000 30     :++
0000 31     : FACILITY:      VAX MACRO ASSEMBLER OBJECT LIBRARY
0000 32     :
0000 33     : ABSTRACT:
0000 34     :
0000 35     : The VAX-11 MACRO assembler translates MACRO-32 source code into object
0000 36     : modules for input to the VAX-11 LINKER.
0000 37     :
0000 38     : ENVIRONMENT:  USER MODE
0000 39     :
0000 40     : AUTHOR: Benn Schreiber, CREATION DATE: 31-AUG-78
0000 41     :
0000 42     : MODIFIED BY:
0000 43     :
0000 44     :      V03-002 MTR0024      Mike Rhodes      4-Feb-1983
0000 45     :      Print the full subtitle text in the Table of Contents.
0000 46     :
0000 47     :      V03.01 MTR0020      Mike Rhodes      7-Jul-1982
0000 48     :      Modify handling of enable/disable/show/noshow directives
0000 49     :      to also respect command qualifier level CLEARING as well
0000 50     :      as SETTING of these options. Modules affected are
0000 51     :      ENABL_DSABL and LIST_NLIST. A new flag MAC$GL_DSLISF is
0000 52     :      employed to facilitate this manipulation.
0000 53     :
0000 54     :      V01.15 RN0029      R. Newland      10-Mar-1980
0000 55     :      Align table of contents so that subtitles containing
0000 56     :      tabs remain correctly aligned.
0000 57     :
```


0000	58	:	V01.14	RN0023	R. Newland	2-Nov-1979
0000	59	:		New message codes to get error message from system		
0000	60	:		message file.		
0000	61	:				
0000	62	:	V01.13	RN0022	R. Newland	31-Oct-1979
0000	63	:		Translate SYS\$LP_LINES to set lines/page		
0000	64	:				
0000	65	:	V01.12	RN0019	R. Newland	25-Oct-1979
0000	66	:		Improve error pointer positioning		
0000	67	:				
0000	68	:	V01.11	RN0005	R. Newland	27-Aug-1979
0000	69	:		Remove .ALIGN LONG statments		
0000	70	:				
0000	71	:	V01.12	RN0009	R. Newland	31-Aug-1979
0000	72	:		Allow maximum size IDENT		
0000	73	:				
0000	74	:	V01.10	;RN0002	R. Newland	01-Feb-1979
0000	75	:		Changes for Source Update Merge, .SBTTL line number		
0000	76	:				
0000	77	:--				


```

0000 79      .SBTTL  DECLARATIONS
0000 80      :
0000 81      : INCLUDE FILES:
0000 82      :
0000 83      :
0000 84      :
0000 85      : MACROS:
0000 86      :
0000 87      :
0000 88      $RABDEF      ;DEFINE RAB OFFSETS
0000 89      $MAC_ADRMODDEF ;DEFINE ADDRESSING MODES
0000 90      $MAC_CTLFLGDEF ;DEFINE CONTROL FLAGS
0000 91      $MAC_INTCODDEF ;DEFINE INT. FILE CODES
0000 92      $MAC_GENVALDEF ;DEFINE COMMON SYMBOLS
0000 93      $MAC_SYMBLKDEF ;DEFINE SYMBOL BLOCK OFFSETS
0000 94      $MACMSGDEF     ; Define message codes
0000 95      DEFSUMCBL      ; Define SUM control block symbols
000U 96
0000 97
00000000 98      .PSECT  MAC$RO_CODE_P1,NOWRT,GBL, LONG

```



```
0000 100 .SBTTL ERRORS FOUND BY THE GRAMMAR/PARSER
0000 101
0000 102 ERRENT:: :ILLEGAL FORMAT FOR .ENTRY
0000 103 $MAC_ERR BADENTRY : Get message code
31 11 0005 104 BRB ERR_0
0007 105
0007 106 ERROPD:: :ILLEGAL FORMAT FOR .OPDEF
0007 107 $MAC_ERR ILLOPDEF : Get message code
2A 11 000C 108 BRB ERR_0
000E 109
000E 110 BDEND1:: :.ENDM DIRECTIVE SEEN
000E 111 :OUTSIDE A MACRO DEFINITION
000E 112 BDEND2:: :.ENDR DIRECTIVE SEEN
000E 113 :OUTSIDE REPEAT BODY
000E 114 $MAC_ERR NOTINMACRO : Get message code
23 11 0013 115 BRB ERR_0 :ISSUE ERROR, SET CR AND RETURN
0015 116
0015 117 ERRMRS:: :MARS TEXT = ERR01
0015 118 $INTOUT X INT$ CHKL :ALIGN LISTING
001B 119 $MAC_ERR UNRECSTMT : Get message code
FFDD' 30 0020 120 BSBW MAC$ERRORPX :ISSUE ERROR TO PASS 2
16 11 0023 121 BRB ERR_1 :JOIN COMMON CODE
0025 122
0025 123 ERRDOL:: :STATEMENT = DIRECTIVE ERR02
0025 124 $MAC_ERR DIRSYNX : Get message code
0C 11 002A 125 BRB ERR_0
002C 126
002C 127 ERRASN:: :ASSIGNMENT = ASSIGN_HEAD ERR02
002C 128 $MAC_ERR ASGNMNTSYN : Get message code
05 11 0031 129 BRB ERR_0
0033 130
0033 131 ERRMST:: :MACHINE_STAT = MACHINE_INST ERR03
0033 132 $MAC_ERR MCHINSTSYN : Get message code
5A FFC5' 30 0038 133 ERR_0: BSBW MAC$ERRORPX : Report error to pass-2
0D 9A 003B 134 ERR_1: MOVZBL #CR,R10 :FORCE READING OF NEW LINE
05 003E 135 RSB
003F 136
003F 137 ERRREF:: :OPERANDS = OPERANDS ERR04
003F 138 :REF = ERR06
003F 139 $MAC_ERR OPRNDSYNX : Get message code
FFB9' 30 0044 140 BSBW MAC$ERRORPX : Issue error to pass-2
FFB6' 31 0047 141 BRW W^MAC$SKP_OPR :SKIP TO NEXT OPERAND FIELD
004A 142
004A 143 ERRIIF:: :IIF_STAT = IIF_HEAD ERR03
004A 144 $MAC_ERR MSGCMAIIF : Get message code
1A 11 004F 145 BRB ERR_3
0051 146
0051 147 ERRDAR:: :DATA_LIST = DATA_ARGS ERR04
0051 148 :DATA_ARGS = DATA_LIST DSQOPN EXPR ERR04
0051 149 $MAC_ERR DATASTSYN : Get message code
13 11 0056 150 BRB ERR_3
0058 151
0058 152 ERRADR:: :ADDR_LIST = ADDR_LIST ERR04
0058 153 :ADDR_STAT = ADDR_TYPE
0058 154 $MAC_ERR ADRLSTSYNX : Get message code
0C 11 005D 155 BRB ERR_3
005F 156
```


		005F	157	ERRCHA::		;CHAR_ARGS = CHAR_ARGS ERR06
		005F	158	\$MAC_ERR ILLASCARG		; Get message code
05	11	0064	159	BRB ERR_3		
		0066	160			
		0066	161	ERRBLK::		;BLOCK_STAT = BLOCK_TYPE ERR03
		0066	162	\$MAC_ERR BLKDIRSYNX		; Get message code
FF92'	30	006B	163	ERR_3: BSBW MAC\$ERRORPT		;ISSUE MESSAGE TO PASS 2
	05	006E	164	RSB		
		006F	165			
		006F	166	ERREXP::		;EXPRESSION ERROR
		006F	167	\$MAC_ERR ILLEXPR		; Get message code
FF89'	30	0074	168	BSBW MAC\$ERRORPT		;ISSUE ERROR TO PASS 2
FF86'	31	0077	169	BRW MAC\$SKP_OPR		;SKIP TO NEXT OPERAND FIELD
		007A	170			
		007A	171	ERRBRF::		;BASIC_REF = DOPN ERR01
		007A	172			;BASIC_REF = DAT RRREG
		007A	173	\$MAC_ERR REGOPSYNX		; Get message code
FF7E'	30	007F	174	BSBW MAC\$ERRORPT		;ISSUE MESSAGE TO PASS 2
0000'CF	06	90	0082	MOVB #ADMS_RRIND,W^MAC\$GB_MODE		;SET MODE TO INDIRECT REGISTER
0000'CF	94	0087	176	CLRB W^MAC\$GB_REG		;USING REGISTER 'R0'
		008B	177	\$INC_PC		;COUNT ONE BYTE
FF6E'	31	008F	178	BRW MAC\$SKP_OPR		;SKIP TO CR OR COMMA AND RETURN


```
0092 180 .SBTTL IDENT PROCESS .IDENT STATEMENT
0092 181
0092 182 :++
0092 183 : FUNCTIONAL DESCRIPTION:
0092 184 :
0092 185 : THIS ROUTINE IS CALLED WHEN A .IDENT IS SCANNED. THE IDENT
0092 186 : IS COPIED INTO THE BUFFER MAC$AB_IDENT.
0092 187 :
0092 188 :--
0092 189
0092 190 IDENT::
56 0000'CF 9E 0092 191 MOVAB W^MAC$AB_IDENT,R6 ;POINT TO IDENT STORAGE
      86 94 0097 192 CLRB (R6)+ ;CLEAR IN CASE NULL IDENT
      FF64' 30 0099 193 BSBW MAC$SKIPSP ;SKIP SPACES
0D 5A 91 009C 194 CMPB R10,#CR ;ARE WE AT END OF LINE?
      49 13 009F 195 BEQL 40$ ;IF EQL YES
7E 5A 90 00A1 196 MOVB R10,-(SP) ;NO--SAVE DELIMITER
55 1F 9A 00A4 197 MOVZBL #SYM$K_MAXLEN,R5 ;SET MAX NUMBER OF CHARACTERS
      00A7 198 :
      00A7 199 : LOOP, COLLECTING IDENT. LOOK FOR END OF LINE OR MATCHING DELIMITER
      00A7 200 :
      FF56' 30 00A7 201 10$: BSBW MAC$GETCHR ;GET NEXT CHARACTER
6E 5A 91 00AA 202 CMPB R10,(SP) ;FIND DELIMITER?
      1C 13 00AD 203 BEQL 20$ ;IF EQL YES
0D 5A 91 00AF 204 CMPB R10,#CR ;NO--END OF LINE?
      17 13 00B2 205 BEQL 20$ ;IF EQL YES
61 8F 5A 91 00B4 206 CMPB R10,#^A/A/+^X20 ;IS CHARACTER LOWER CASE?
      09 1F 00B8 207 BLSSU 15$ ;IF LSSU NO
7A 8F 5A 91 00BA 208 CMPB R10,#^A/Z/+^X20 ;MAYBE...
      03 1A 00BE 209 BGTRU 15$ ;IF GTRU NO
      5A 20 8A 00C0 210 BICB #^X20,R10 ;YES--MAKE UPPER CASE
86 5A 90 00C3 211 15$: MOVB R10,(R6)+ ;NO--STORE CHARACTER
DE 55 F4 00C6 212 SOBGEQ R5,10$ ; Loop if there is room
      55 D7 00C9 213 DECL R5 ;MAKE R5 NEGATIVE FOR IDENT TOO LONG
50 1F 55 C3 00CB 214 20$: SUBL3 R5,#SYM$K_MAXLEN,R0 ;FIGURE LENGTH OF IDENT
0000'CF 50 90 00CF 215 MOVB R0,W^MAC$AB_IDENT ;STORE AS FIRST BYTE OF IDENT
      8E 5A 91 00D4 216 CMPB R10,(SP)+ ;END WITH DELIMITER?
      11 13 00D7 217 BEQL 40$ ;IF EQL YES
      00D9 218 $MAC_ERR UNTERMARG ; No--assume unterminated arg
      55 D5 00DE 219 TSTL R5 ;BUT CHECK TO SEE
      05 18 00E0 220 BGEQ 30$ ;IF GEQ UNTERM. ARG
      00E2 221 $MAC_ERR ILLSYMLEN ; else IDENT is too long
      FF16' 30 00E7 222 30$: BSBW MAC$ERRORPT ;REPORT ERROR
5A 0D 9A 00EA 223 40$: MOVZBL #CR,R10 ;FORCE READING OF NEW LINE
      05 00ED 224 RSB
```



```
00EE 226 .SBTTL TITLE PROCESS .TITLE STATEMENT
00EE 227
00EE 228 :++
00EE 229 : FUNCTIONAL DESCRIPTION:
00EE 230 :
00EE 231 : THIS ROUTINE IS CALLED WHEN A .TITLE DIRECTIVE IS SCANNED.
00EE 232 : THE REST OF THE SOURCE LINE IS READ AND THE FIRST WORD IS
00EE 233 : STORED AS THE PROGRAM TITLE, AND THE REST IS STORED AS THE
00EE 234 : TITLE SUB-COMMENT IN THE LISTING HEADER BUFFER.
00EE 235 :
00EE 236 :--
00EE 237
00EE 238 TITLE::
00EE 239 BSBW MAC$SYMSCNUP ;SCAN THE SYMBOL
56 0000'CF 9E 00F1 240 BLBC R0,40$ ;BRANCH IF NO TITLE SCANNED
55 66 9A 00F4 241 MOVAB W^MAC$AB_TMSYM,R6 ;POINT TO TEMP SYMBOL NAME BLOCK
20 66 55 D6 00FC 242 MOVZBL (R6),R5 ;GET LENGTH OF SYMBOL
0000'CF 20 2C 00FE 243 INCL R5 ;COPY THE BYTE COUNT ALSO
1F 20 66 55 2C 0102 244 MOVCS R5,(R6),#^A/ /,- ;COPY INTO BUFFER WITH BLANK PADDING
0000'CF 55 86 9A 0106 245 MOVZBL (R6)+,R5 ;GET TITLE LENGTH AGAIN
20 66 55 2C 0109 246 MOVCS R5,(R6),#^A/ /,#SYM$K_MAXLEN,- ;COPY INTO PAGE HEADER BUFFER
0000'CF 010E 247 W^MAC$AB_HD_TITLE
56 0000'CF 9E 0111 248 BSBW MAC$SKIPSP ;SKIP SPACES
20 6E 00 2C 0114 249 MOVAB W^MAC$AB_HD_TSTRG,R6 ;POINT TO WHERE TITLE SUBSTRING GOES
66 28 011D 250 MOVCS #0,(SP),#^A7 /,- ;BLANK FILL THE TITLE BUFFER
55 00000001'8F 0000'CF C1 011F 251 #LST$K_TITLE_SIZ,(R6) ;...
55 0000'CF C2 0129 252 W^MAC$GL_LINELN,#MAC$AB_LINEBF+1,R5 ;COMPUTE LENGTH OF SUBSTRING
22 15 012E 253 SUBL2 W^MAC$GL_LINEPT,R5 ;...
28 55 91 0130 254 BLEQ 40$ ;IF LEQ NO SUBSTRING
55 03 1B 0133 255 CMPB R5,#LST$K_TITLE_SIZ ;STRING TOO LONG?
0000'CF 55 28 9A 0135 256 BLEQU 10$ ;IF LEQ NO
54 0000'CF 01 C3 0138 257 MOVZBL #LST$K_TITLE_SIZ,R5 ;YES--USE MAXIMUM SIZE
0000'CF 01 C3 013D 258 MOVL R5,W^MAC$GL_TTX_SIZ ;SAVE SUBSTRING LENGTH
10$: 259 SUBL3 #1,W^MAC$GL_LINEPT,R4 ;POINT TO BEGINNING OF TITLE SUBSTRING
0143 260
0143 261 : COPY TITLE SUBSTRING INTO PAGE HEADER BUFFER
0143 262
0143 263
66 84 90 0143 264 20$: MOVB (R4)+,(R6) ;COPY A BYTE
09 86 91 0146 265 CMPB (R6)+,#TAB ;IS CHARACTER A TAB?
04 12 0149 266 BNEQ 30$ ;IF NEQ NO
FF A6 20 90 014B 267 MOVB #^A/ /,-1(R6) ;YES--MAKE INTO A SPACE
F1 55 F5 014F 268 30$: SOBGTR R5,20$ ;DO WHOLE STRING
5A 0D 9A 0152 269 40$: MOVZBL #CR,R10 ;FORCE READING OF NEW LINE
05 0155 270 RSB
```



```
0156 272 .SBTTL SBTTL PROCESS .SBTTL STATEMENT
0156 273
0156 274 :++
0156 275 : FUNCTIONAL DESCRIPTION:
0156 276 :
0156 277 : THIS ROUTINE PROCESSES THE .SBTTL STATEMENT. THE SUBTITLE
0156 278 : LINE IS READ AND WRITTEN TO THE INTERMEDIATE FILE. IF WE
0156 279 : ARE LISTING, THE SUBTITLE LINE IS OUTPUT TO THE LISTING FILE
0156 280 :
0156 281 :--
0156 282
0156 283 SBTTL::
56 0000'CF 00000001'8F DD 0156 284 PUSHL R8 ;PRESERVE R8
56 0000'CF 58 C1 0158 285 ADDL3 #MAC$AB_LINEBF+1,W^MAC$GL_LINELN,R6 ;FIGURE LENGTH OF SUBTITLE
58 56 C2 0162 286 SUBL2 W^MAC$GL_LINEPT,R6
28 56 D0 0167 287 MOVL R6,R8 ;SAVE LENGTH FOR TABLE OF CONTENTS.
56 28 91 016A 288 CMPB R6,#LST$K_TITLE_SIZ ;IS SIZE OK?
03 1B 016D 289 BLEQU 10$ ;IF LEQ YES
56 28 9A 016F 290 MOVZBL #LST$K_TITLE_SIZ,R6 ;NO--USE MAXIMUM
50 59 56 C1 0172 291 10$: ADDL3 R6,R9,R0 ;SEE IF THERE IS ENOUGH ROOM IN THE BUFFER
50 50 03 C0 0176 292 ADDL2 #3,R0 ;COUNT THE 2 COUNT BYTES AND ACTION BYTE
0000'CF 50 D1 0179 293 CMPL R0,W^MAC$GL_INTWRNPT ;ROOM IN THE BUFFER?
03 1B 017E 294 BLEQU 20$ ;IF LEQ YES
50 03 FE7D' 30 0180 295 BSBW MAC$OUTFRAME ;NO--DUMP THE BUFFER
50 03 FE76' 30 0183 296 20$: ADDL3 R6,#3,R0 ;FIGURE TOTAL LENGTH OF FRAME
89 21 90 018A 297 BSBW MAC$INTOUT_N ;SET TO STORE SBTTL IN BUFFER
89 56 90 018D 298 MOVB #INT$ SBTTL,(R9)+ ;STORE THE ACTION CODE
03 14 0190 299 MOVB R6,(R9)+ ;STORE LENGTH OF SBTTL STRING
00CC 31 0192 300 BGTR 40$ ;IF GTR THERE IS A SUBTITLE STRING
55 0000'CF 01 C3 0195 301 30$: BRW 100$ ;ELSE WE ARE DONE NOW
55 DD 0198 302 40$: SUBL3 #1,W^MAC$GL_LINEPT,R5 ;POINT TO REAL START OF SBTTL STRING
59 DD 019D 303 PUSHL R5 ;SAVE OVER MOVC
69 65 56 28 019F 304 MOVCL3 R6,(R5),(R9) ;SAVE POINTER INTO INTERMEDIATE BUFFER
59 53 D0 01A3 305 MOVCL3 R6,(R5),(R9) ;COPY STRING TO INTERMEDIATE BUFFER
52 8ED0 01A6 306 MOVL R3,R9 ;UPDATE POINTER INTO INTERMEDIATE BUFFER
62 56 09 3A 01A9 307 POPL R2 ;GET POINTER TO START OF SUBTTL IN BUFFER
62 05 13 01AD 308 50$: LOCC #TAB,R6,(R2) ;FIND ANY TABS IN THE SUBTITLE IN INT. BUFFE
61 20 90 01AF 309 BEQL 60$ ;IF EQL NO MORE
F5 11 01B2 310 MOVB #^A/ /,(R1) ;FOUND ONE--CHANGE TO SPACE
0000'CF 65 58 28 01B4 311 BRB 50$ ;LOOK FOR MORE
D1 68 09 E1 01BD 312 60$: POPL R5 ;RESTORE POINTER TO SBTTL STRING
31 68 13 E2 01C1 313 MOVCL3 R8,(R5),W^MAC$AB_LINEBF ;COPY FOR TABLE OF CONTENTS
01C5 314 BBC #FLG$V_LSTXST,(R11),30$ ;BRANCH IF NOT DOING LISTINGS
01C5 315 BBSS #FLG$V_TOCLFG,(R11),70$ ;SET TOC FLAG AND SEE IF WE NEED A HEADER
01C5 316 :
01C5 317 : OUTPUT TABLE OF CONTENTS HEADER
01C5 318 :
0000'CF 0000'CF D0 01C5 319 MOVL W^MAC$GL_LN_PAGE,W^MAC$GL_LINE_CNT ; Set # lines left on page
55 0000'CF 9E 01CC 320 MOVAB W^MAC$LIST_RAB,R5 ;POINT TO LISTING RAB
22 A5 0001'8F B0 01D1 321 MOVW #MAC$K_HD_SIZE+1,RAB$W_RSZ(R5) ;SET THE RECORD SIZE
28 A5 0000'CF 9E 01D7 322 MOVAB W^MAC$AB_HD_NEWPG,RAB$C_RBF(R5) ;AND THE RECORD ADDRESS
0088 30 01DD 323 BSBW SBT PUT CIN ;WRITE THE TITLE LINE
50 0000'CF 9E 01E0 324 MOVAB W^MAC$AB_TOC_MSG,R0 ;POINT TO TABLE OF CONTENTS MSG
22 A5 80 9B 01E5 325 MOVZBW (R0)+,RAB$W_RSZ(R5) ;SET THE RECORD SIZE
28 A5 50 D0 01E9 326 MOVL R0,RAB$L_RBF(R5) ;AND THE RECORD ADDRESS
0078 30 01ED 327 BSBW SBT PUT CIN ;WRITE THE LINE
22 A5 B4 01F0 328 CLRW RAB$W_RSZ(R5) ;WRITE A BLANK LINE
```



```

      0072 30 01F3 329 BSBW SBT PUT LIN ;WRITE THE BLANK LINE
81 51 FFE8'CF 9E 01F6 330 70$: MOVAB W*MAC$AB_LINEBF-24,R1 ; Point into listing buffer
      2020 8F D0 01FB 331 MOVL #A/ 7,(R1)+ ; Store four spaces
      81 28 90 0202 332 MOVB #A/(/,(R1)+ ; and left parenthesis
50 0000'CF D0 0205 333 MOVL W*MAC$GL_SRCFAG,R0 ;GET THE SOURCE PAGE NUMBER
      FDF3' 30 020A 334 BSBW MAC$DEC_OUT_L2X ;OUTPUT THE PAGE NUMBER
80 2029 8F B0 020D 335 MOVB #A/ /,(R0)+ ;STORE RIGHT PAREN-SPACE
      80 20 90 0212 336 80$: MOVB #A/ /,(R0)+ ;PAD WITH SPACES
0000'8F 50 B1 0215 337 CMPW R0,#MAC$AB_LINEBF ;DONE PADDING?
      F6 1F 021A 338 BLSSU 80$ ;LOOP FOR ALL
51 50 08 C3 021C 339 SUBL3 #8,R0,R1 ; Form pointer to line number
0000'CF 51 D0 0220 340 MOVL R1,W*MAC$GL_LIST_PTR ; Set up pointer
50 0000'CF 3C 0225 341 MOVZWL W*MAC$GW_LST_LINE,R0 ; Get listing line number
      FDD3' 30 022A 342 BSBW MAC$DEC_OUT_R2L ;OUTPUT PAGE NUMBER
      11 6B 27 E1 022D 343 BBC #FLG$V_OPDFIL,(R11),90$ ; Branch if file is not being updated
001C'CF 02 E1 0231 344 BBC #SUM_V_SRCUPD,- ; Branch if line is from source
      0B 90 0236 345 W*MAC$GT_SCB+SUM_B_FLAGS,90$
      81 2E 90 0237 346 MOVB #A/./,(R1)+ ; Store period
50 0000'CF 3C 023A 347 MOVZWL W*MAC$GW_LST_INST,R0 ; Get insert number
      FDBE' 30 023F 348 BSBW MAC$DEC_OUT_L2X ; Output the insert number
      0242 349 90$: ADDL2 #24,R8 ; Count the line/page
55 58 18 C0 0242 350 MOVAB W*MAC$LIST_RAB,R5 ;POINT TO LISTING RAB
      0000'CF 9E 0245 351 MOVL R8,RAB$W_RSZ(R5) ;STORE RECORD SIZE
28 22 A5 58 D0 024A 352 MOVAB W*MAC$AB_LINEBF-24,RAB$L_RBF(R5) ; and the record address
      0011 30 0254 353 BSBW SBT PUT LIN ;WRITE THE TOC LINE
      0000'CF D7 0257 354 DECL W*MAC$GL_LINE_CNT ;ONE LESS LINE TO GO
      04 12 025B 355 BNEQ 100$ ;IF NEQ MORE ROOM ON PAGE
      00 6B 13 E5 025D 356 BBCC #FLG$V_TOCFLG,(R11),100$ ;ELSE FLAG NEW HEADER NEEDED
      58 8ED0 0261 357 POPL R8 ;RESTORE R8.
      5A 0D 9A 0264 358 100$: MOVZBL #CR,R10 ;FORCE READING OF NEW LINE
      05 0267 359 RSB
      0268 360
      0268 361 SBT_PUT_LIN:
13 6B 09 E1 0268 362 BBC #FLG$V_LSTXST,(R11),10$ ;BRANCH IF LISTING DISABLED
      026C 363 $PUT RAB=(R5),- ;WRITE LINE TO LISTING FILE
      026C 364 ERR=W*MAC$ERR_PUT ;REPORT ANY ERRORS
      03 50 E8 0279 365 BLBS R0,10$ ;BRANCH IF GOOD PUT
      FD81' 30 027C 366 BSBW MAC$CLOSE_LIST ;ELSE CLOSE THE LISTING FILE
      05 027F 367 RSB ;KEEP ASSEMBLING
      10$: 368
```



```
0280 370 .SBTTL ENABL/DSABL PROCESS .ENABL/.DSABL
0280 371
0280 372 :++
0280 373 : FUNCTIONAL DESCRIPTION:
0280 374 :
0280 375 : THESE TWO ROUTINES PROCESS .ENABL/.DSABL DIRECTIVES. THE
0280 376 : APPROPRIATE FLAGS ARE SET/CLEARED ON PASS 1 AND CODE IS
0280 377 : PUT IN THE INTERMEDIATE BUFFER TO DO THE SAME ON PASS 2.
0280 378 :
0280 379 :--
0280 380
0280 381 ENABL::
50 FF 8F 98 0280 382 CVTBL #-1,R0 ;DIRECTIVE = KENABL
0280 383 BRB ENABL_DSABL ;ENABLE ANY OPTIONS
0286 384 ;GO TO COMMON CODE
0286 385 DSABL::
50 D4 0286 386 CLRL R0 ;DIRECTIVE = KDSABL
0288 387 ;DISABLE ANY OPTIONS
0288 388 ENABL_DSABL:
0288 389
0000'CF 50 D0 0288 390 MOVL R0,W^MAC$GL DIRFLG ;SET THE FLAG FOR LATER
FD70' 30 028D 391 BSBW MAC$SYMSCNUP ;SCAN NEXT OPTION
67 50 E9 0290 392 BLBC R0,50$ ;BRANCH IF NO OPTION SCANNED
55 0000'CF 9E 0293 393 MOVAB W^ENB$G_OPTIONS,R5 ;POINT TO ENABLE OPTIONS LIST
FD65' 30 0298 394 BSBW MAC$SRC_LIST ;LOOK FOR THE OPTION WE SCANNED
1B 50 E8 029B 395 BLBS R0,20$ ;BRANCH IF FOUND
55 0000'CF 9E 029E 396 MOVAB W^ENB$G_LONGNAMES,R5 ;NO--TRY THE LONG NAMES
FD5A' 30 02A3 397 BSBW MAC$SRC_LIST ;LOOK FOR IT
06 50 E9 02A6 398 BLBC R0,10$ ;BRANCH IF NOT FOUND
51 05 A1 D0 02A9 399 MOVL SYMSL_VAL(R1),R1 ;POINT TO THE REAL BLOCK
0A 11 02AD 400 BRB 20$ ;AND CONTINUE
FD49' 30 02AF 401 10$: $MAC_ERR NOTENABOPT ;Get error message code
34 11 02B4 402 BSBW MAC$ERRORLN ;SEND TO PASS 2
0000'8F 51 B1 02B7 403 BRB 40$
08 12 02B9 404 20$: CMPW R1,#ENB$G_LOCALSYMB ;WAS THIS '.ENABL LSB'?
03 0000'CF E9 02BE 405 BNEQ 30$ ;IF NEQ NO
FD38' 30 02C0 406 BLBC W^MAC$GL DIRFLG,30$ ;YES--BRANCH IF DISABLE
0000'CF 09 A1 B3 02C5 407 BSBW MAC$SET_NEW_LSB ;NO--ENABLE A NEW LSB
1D 12 02C8 408 30$: BITW SYMSW_FLAG(R1),W^MAC$GL_ENLISF ;SET BY COMMAND?
0000'CF 09 A1 B3 02CE 409 BNEQ 40$ ;IF NEQ YES--DO NOT CHANGE HERE
15 12 02D0 410 BITW SYMSW_FLAG(R1),W^MAC$GL_DSLISF ;CLEARED BY COMMAND?
52 05 A1 9E 02D6 411 BNEQ 40$ ;IF NEQ YES--DO NOT CHANGE HERE
62 0000'CF D0 02D8 412 MOVAB SYMSL_VAL(R1),R2 ;POINT TO THE VALUE WORD FOR OPTION
FD10' 30 02DC 413 MOVL W^MAC$GL DIRFLG,(R2) ;SET/CLEAR THE OPTION
2C 5A 91 02E1 414 $INTOUT_LW INT$ SETLONG,<W^MAC$GL DIRFLG,R2> ;SET/CLEAR FLAG ON PASS 2
93 12 02ED 415 40$: BSBW MAC$SKIPSP ;SKIP SPACES
FD08' 30 02F0 416 CMPB R10,#^A/,/ ;SCAN TO A COMMA?
F3 11 02F3 417 BNEQ ENABL_DSABL ;IF NEQ NO--SCAN FOR NEXT OPTION
05 02F5 418 BSBW MAC$GETCHR ;YES--SKIP THE COMMA
02F8 419 BRB 40$ ;CONTINUE
02FA 420 50$: RSB
```



```
02FB 422 .SBTTL LIST/NLIST PROCESS .LIST/.NLIST
02FB 423
02FB 424 :++
02FB 425 : FUNCTIONAL DESCRIPTION:
02FB 426 :
02FB 427 : THESE TWO ROUTINES PROCESS THE .LIST/.NLIST DIRECTIVES.
02FB 428 : THE LINE IS SCANNED TO GET THE OPTIONS (IF ANY) AND THE
02FB 429 : APPROPRIATE FLAGS ARE CLEARED IN PASS 1 AND CODE IS SENT
02FB 430 : TO THE INTERMEDIATE BUFFER TO DO SO ON PASS 2.
02FB 431 :
02FB 432 :--
02FB 433
02FB 434 LIST:: ;DIRECTIVE = KLIST
50 FF 8F 98 02FB 435 CVTBL #-1,R0 ;SET ANY FLAGS
01 DD 02FF 436 PUSHL #1 ;INCREMENT LISTING LEVEL
06 11 0301 437 BRB LIST_NLIST
0303 438
0303 439 NLIST:: ;DIRECTIVE = KNLIST
7E FF 50 D4 0303 440 CLRL R0 ;CLEAR ANY FLAGS
8F 98 0305 441 CVTBL #-1,-(SP) ;DECREMENT LEVEL
0309 442
0309 443 LIST_NLIST:
0309 444
0000'CF 50 D0 0309 445 MOVL R0,W^MAC$GL_DIRFLG ;SET THE FLAG FOR LATER
FCF' 30 030E 446 BSBW MAC$SYMSCNUP ;SCAN FOR AN OPTION
24 50 E8 0311 447 BLBS R0,10$ ;BRANCH IF OPTION SCANNED
0000'CF 8E C0 0314 448 ADDL2 (SP)+,W^MAC$GL_LIST_LVL ;NO--ADJUST LISTING LEVEL
0319 449 $INTOUT_LW INT$ SETLONG,- ;SEND CODE FOR PASS 2 TO DO THE SAME
0319 450 <W^MAC$GL_LIST_LVL,#MAC$GL_LIST_LVL>
0329 451 $INTOUT_LW INT$ SETLONG,<#1,#MAC$GL_LIST_IT> ;SET '.LIST/.NLIST' FLAG
05 0337 452 RSB ;ALL DONE
0338 453
0338 454 : THERE WAS AT LEAST ONE OPTION ON THE LINE
0338 455
0338 456 10$: TSTL (SP)+ ;CLEAR THE STACK
55 0000'CF 9E 033A 457 MOVAB W^LST$G_DIRLIST,R5 ;POINT TO OPTION NAMES
FCBE' 30 033F 458 BSBW MAC$SRC_LIST ;LOOK UP THE OPTION NAME
1B 50 E8 0342 459 BLBS R0,40$ ;BRANCH IF FOUND
55 0000'CF 9E 0345 460 MOVAB W^LST$G_LONGNAMES,R5 ;NO--TRY THE LONG NAMES
FCB3' 30 034A 461 BSBW MAC$SRC_LIST ;LOOK FOR IT
06 50 E9 034D 462 BLBC R0,30$ ;BRANCH IF NOT FOUND
51 05 A1 D0 0350 463 MOVL SYM$L_VAL(R1),R1 ;POINT TO THE REAL BLOCK
0A 11 0354 464 BRB 40$ ;AND CONTINUE
FCA2' 30 0356 465 30$: $MAC_ERR NOTLGLISOP ;Get message code
23 11 035B 466 BSBW MAC$ERRORLN ;SEND TO PASS 2
035E 467 BRB 50$
0360 468
0360 469 : OPTION WAS FOUND
0360 470
0000'CF 09 A1 B3 0360 471 40$: BITW SYM$W_FLAG(R1),W^MAC$GL_ENLISF ;SET BY COMMAND?
1B 12 0366 472 BNEQ 50$ ;YES--DO NOT CHANGE HERE
0000'CF 09 A1 B3 0368 473 BITW SYM$W_FLAG(R1),W^MAC$GL_DSLISF ;CLEARED BY COMMAND?
13 12 036E 474 BNEQ 50$ ;IF NEQ YES--DO NOT CHANGE HERE
52 05 A1 9E 0370 475 MOVAB SYM$L_VAL(R1),R2 ;POINT TO VALUE
62 0000'CF D0 0374 476 MOVL W^MAC$GL_DIRFLG,(R2) ;SET/CLEAR FLAG
FC7A' 30 0379 477 $INTOUT_LW INT$ SETLONG,<(R2),R2> ;TELL PASS 2 TO DO IT TOO
0383 478 50$: BSBW MAC$SKIPSP ;SKIP SPACES
```


2C	5A	91	0386	479	CMPB	R10,#^A/,/	;DID WE GET TO A COMMA?
	05	12	0389	480	BNEQ	60\$;IF NEQ NO
	FC72'	30	038B	481	BSBW	MAC\$GETCHR	;YES--GET NEXT CHARACTER
	F3	11	038E	482	BRB	50\$:
	FC6D'	30	0390	483	BSBW	MAC\$SYMSCNUP	;FIND AN OPTION
A4	50	E8	0393	484	BLBS	RO,20\$;BRANCH IF OPTION SCANNED
		05	0396	485	RSB		


```
0397 487 .SBTTL PROCESS .CROSS/.NOCROSS DIRECTIVES
0397 488
0397 489 :++
0397 490 : FUNCTIONAL DESCRIPTION:
0397 491 :
0397 492 : THESE TWO ROUTINES PROCESS THE .CROSS AND .NOCROSS
0397 493 : DIRECTIVES.
0397 494 :
0397 495 :--
0397 496
0397 497 CROENB:: ;DIRECTIVE = KCROSS
03F2'CF 9F 0397 498 PUSHAB W^CLR_XCRF ;ACTION ROUTINE FOR SYMBOLS
56 D4 039B 499 CLRL R6 ;FLAG THIS IS ENABLE
07 11 039D 500 BRB CROS_0 ;JOIN COMMON CODE
039F 501
039F 502 CRODSB:: ;DIRECTIVE = KNCROS
03EC'CF 9F 039F 503 PUSHAB W^SET_XCRF ;ACTION ROUTINE FOR SYMBOLS
56 01 D0 03A3 504 MOVL #1,R6 ;FLAG THIS IS DISABLE
FC57' 30 03A6 505 CROS_0: BSBW MAC$SKIPSP ;SKIP SPACES
OD 5A 91 03A9 506 CMPB R10,#CR ;JUST THE DIRECTIVE?
2F 13 03AC 507 BEQL 70$ ;IF EQL YES
03AE 508 :
03AE 509 : LOOP, SCANNING SYMBOL NAMES. SET OR CLEAR SYM$M_XCRF IN THE
03AE 510 : SYMBOL FLAGS FOR EACH SYMBOL FOUND, AS APPROPRIATE.
03AE 511 :
FC4F' 30 03AE 512 20$: BSBW MAC$SYMSCNUP ;SCAN A SYMBOL NAME
OA 50 E8 03B1 513 BLES R0,30$ ;BRANCH IF WE GOT ONE
03B4 514 $MAC_ERR DIRSYNX ; No--syntax error
FC44' 30 03B9 515 BSBW MAC$ERRORPT ;REPORT THE ERROR
1C 11 03BC 516 BRB 60$ ;FINISH UP
FC3F' 30 03BE 517 30$: BSBW MAC$INSUSRSYMTB ;LOOK UP THE SYMBOL NAME (INSRT IF NOT FOUND
03 50 E9 03C1 518 BLBC R0,40$ ;IF NOT FOUND, JUST IGNORE IT
00 BE 16 03C4 519 JSB @($P) ;CALL ACTION ROUTINE TO SET/CLEAR XCRF
FC36' 30 03C7 520 40$: BSBW MAC$SKIPSP ;SKIP SPACES
2C 5A 91 03CA 521 CMPB R10,#^A/. / ;STOP ON A COMMA?
06 12 03CD 522 BNEQ 50$ ;IF NEQ NO
FC2E' 30 03CF 523 BSBW MAC$GETCHR ;YES--GET NEXT CHAR
FC2B' 30 03D2 524 BSBW MAC$SKIPSP ;AND THEN SKIP SPACES
OD 5A 91 03D5 525 50$: CMPB R10,#CR ;END OF LINE?
D4 12 03D8 526 BNEQ 20$ ;IF NEQ NO
8E D5 03DA 527 60$: TSTL (SP)+ ;YES--CLEAR STACK
05 03DC 528 RSB
03DD 529 :
03DD 530 : THERE WERE NO SYMBOL NAMES ON THE LINE. IF .CROSS, CLEAR THE XCRF
03DD 531 : BIT IN FLAGS. IF .NOCROSS, SET IT.
03DD 532 :
05 8E D5 03DD 533 70$: TSTL (SP)+ ;CLEAR THE STACK
00 05 56 E9 03DF 534 BLBC R6,90$ ;BRANCH IF THIS WAS .CROSS
00 6B 1F E3 03E2 535 BBCS #FLG$V_XCRF,(R11),.+1 ;.NOCROSS--SET XCRF FLAG
05 03E6 536 RSB ;ALL DONE
00 6B 1F E5 03E7 537 90$: BBCC #FLG$V_XCRF,(R11),.+1 ;.CROSS--CLEAR XCRF FLAG
05 03EB 538 RSB
03EC 539
03EC 540 .DEBUG SET_XCRF, CLR_XCRF
03EC 541
03EC 542 SET_XCRF:
00 09 A1 0C E3 03EC 543 BBCS #SYM$V_XCRF,SYM$W_FLAG(R1),.+1 ;DISABLE CREF FOR THIS SYMBOL
```


00	09	A1	0C	E5	03F1	544	RSB	
				05	03F2	545		
					03F2	546	CLR_XCRF:	
				E5	03F2	547	BBCC	#SYMSV_XCRF,SYMSW_FLAG(R1),.+1 ;ENABLE CREF FOR THIS YMBOL
				05	03F7	548	RSB	


```
03F8 550      .SBTTL SETDFL PROCESS .DEFAULT DIRECTIVE
03F8 551
03F8 552      :++
03F8 553      : FUNCTIONAL DESCRIPTION:
03F8 554      :
03F8 555      : THIS ROUTINE PROCESSES THE .DEFAULT DIRECTIVE. THE ONLY
03F8 556      : ARGUMENT CURRENTLY IMPLEMENTED IS 'DISPLACEMENT'. THIS
03F8 557      : SETS THE DEFAULT DISPLACEMENT TO USE IN PC-RELATIVE EXPRESSIONS
03F8 558      : WHEN NO EXPLICIT DISPLACEMENT IS SPECIFIED.
03F8 559      :
03F8 560      :--
03F8 561
03F8 562      SETDFL::
03F8 563      BSBW      MAC$SYMSCNUP      ;DIRECTIVE = KDFLT
03F8 564      BLBS      R0,10$      ;GET THE THING WE ARE DEFAULTING
03FE 565      $MAC_ERR DIRSYNX      ;BRANCH IF WE FOUND SOMETHING
55 0000'CF 11 0403 566      BRB      60$      ;No symbol--that's an error
07 50 E8 0405 567 10$: MOVAB W^MAC$DFLT_LIST,R5 ;REPORT ERROR AND RETURN
07 50 E8 040A 568      BSBW      MAC$SRC_LIST ;POINT TO THE LIST OF VALID ARGS
07 50 E8 040D 569      BLBS      R0,20$ ;LOOK UP THE ARG
07 50 E8 0410 570      $MAC_ERR ILLDFLTARG ;BRANCH IF FOUND
07 50 E8 0415 571      BRB      60$ ;No--report the error
07 50 E8 0417 572      : ;AND RETURN
07 50 E8 0417 573      : THERE IS CURRENTLY ONLY ONE THING DEFAULTED. HENCE, IT IS
07 50 E8 0417 574      : NOT CHECKED.
07 50 E8 0417 575      :
07 50 E8 0417 576 20$: BSBW      MAC$SKIPSP      ;SKIP SPACES
07 50 E8 041A 577      CMPB      R10,#^A/,/ ;SKIP TO A COMMA?
07 50 E8 041D 578      BNEQ      30$ ;IF NEQ NO
07 50 E8 041F 579      BSBW      MAC$GETCHR ;YES--SKIP IT
07 50 E8 0422 580      BSBW      MAC$SKIPSP ;THEN SKIP SPACES
07 50 E8 0425 581 30$: BSBW      MAC$SYMSCNUP ;SCAN FOR ANOTHER SYMBOL
07 50 E8 0428 582      BLBS      R0,40$ ;BRANCH IF WE FOUND ONE
07 50 E8 042B 583      $MAC_ERR DIRSYNX ;No--syntax error
07 50 E8 0430 584      BRB      60$ ;REPORT ERROR
55 0000'CF 11 0432 585 40$: MOVAB W^MAC$DSPL_ARGS,R5 ;POINT TO DISPLACEMENT ARGS LIST
07 50 E8 0437 586      BSBW      MAC$SRC_LIST ;LOOK UP THE OPTION
07 50 E8 043A 587      BLBS      R0,50$ ;BRANCH IF FOUND
07 50 E8 043D 588      $MAC_ERR ILLDFLTARG ;No--illegal arg
07 50 E8 0442 589      BRB      60$ ;REPORT ERROR
0000'CF 05 A1 D0 0444 590 50$: MOVL SYM$ VAL(R1),W^MAC$GL_D ;PC DSP ;SET DEFAULT DISPLACEMENT
07 50 E8 044A 591      BSBW      MAC$SKIPSP ;NOW SKIP SPACES
07 50 E8 044D 592      CMPB      R10,#CR ;MAKE SURE WE FOUND END OF LINE
07 50 E8 0450 593      BEQL      70$ ;IF EQL WE FOUND IT
07 50 E8 0452 594      $MAC_ERR DIRSYNX ;OOPS--AND WE WERE DOING SO WELL
07 50 E8 0457 595 60$: BSBW      MAC$ERRORLN ;REPORT THE ERROR
07 50 E8 045A 596      MOVL      #CR,R10 ;FORCE END OF LINE
07 50 E8 045D 597 70$: RSB
```



```
045E 599 .SBTTL ENDPRG PROCESS .END STATEMENT
045E 600
045E 601 :++
045E 602 : FUNCTIONAL DESCRIPTION:
045E 603 :
045E 604 : THIS ROUTINE IS CALLED WHEN THE '.END' STATEMENT IS ENCOUNTERED.
045E 605 : IT WILL SCAN FOR A TRANSFER ADDRESS AND STORE IT AWAY IF IT
045E 606 : IS PRESENT. A JUMP IS THEN EXECUTED TO END PASS 1.
045E 607 :
045E 608 :--
045E 609
045E 610 ENDPRG::
045E 611 BSBW MAC$SYMSCNUP ;DIRECTIVE = KEND
OC 50 E8 0461 612 BLBS R0,10$ ;SCAN FOR A SYMBOL NAME
OD 5A 91 0464 613 CMPB R10,#CR ;BRANCH IF SYMBOL SCANNED
1C 13 0467 614 BEQL 40$ ;NO--BUT DID WE FIND END OF LINE?
0469 615 $MAC_ERR DIRSYNX ;IF EQL YES--OK
046E 616 BRB 20$ ;Else issue syntax error
FB8D' 30 0470 617 10$: BSBW MAC$SRCUSRSYMTB ;ISSUE MESSAGE
OA 50 E8 0473 618 BLBS R0,30$ ;YES--LOOK UP NAME
0476 619 $MAC_ERR UNDEFXFRAD ;IF LBS THEN NAME FOUND OK
FB82' 30 047B 620 20$: BSBW MAC$ERRORLN ;Undefined--set to issue message
05 11 047E 621 BRB 40$ ;ISSUE MESSAGE TO PASS 2
0000'CF 51 D0 0480 622 30$: MOVL R1,W^MAC$GL_XFRADR ;SET POINTER TO SYMBOL BLOCK
0000'CF D5 0485 623 40$: TSTL W^MAC$GL_IF_LEVEL ;ANY OUTSTANDING CONDITIONALS?
08 15 0489 624 BLEQ 50$ ;IF LEQ NO
048B 625 $MAC_ERR UNTERMCOND ;Yes--get message code
FB6D' 30 0490 626 BSBW MAC$ERRORLN ;SEND MESSAGE TO PASS 2
FB6A' 30 0493 627 50$: BSBW MAC$SET_PC ;RECORD PC
FB67' 31 0496 628 BRW W^MAC$PASS1_END ;FINISH PASS 1
0499 629
0499 630 .END
```


\$\$TMP1	=	00000002		
\$\$TMP2	=	00000065		
\$COUNT	=	0000003B		
ADMS_ABSOLUTE	=	00000002		
ADMS_BYTE_DISP	=	0000000A		
ADMS_DFBYTEDISP	=	0000000B		
ADMS_DFLONGDISP	=	0000000F		
ADMS_DFRAUTOINC	=	00000009		
ADMS_DFWORDDISP	=	0000000D		
ADMS_IMMEDIATE	=	00000001		
ADMS_INDEX	=	00000004		
ADMS_LITERAL	=	00000000		
ADMS_LONG_DISP	=	0000000E		
ADMS_MAXMOD	=	0000000F		
ADMS_PIC	=	00000003		
ADMS_REGAUTODEC	=	00000007		
ADMS_REGAUTOINC	=	00000008		
ADMS_REGISTER	=	00000005		
ADMS_RRIND	=	00000006		
ADMS_WORD_DISP	=	0000000C		
ARG\$K_SIZE	=	000003E8		
AUD\$K_SIZE	=	00000010		
BDEND1		0000000E	RG	03
BDEND2		0000000E	RG	03
BIT...	=	00000005		
BLNK	=	00000020		
CHR\$M_COMMA CR	=	00000020		
CHR\$M_ILL CHR	=	00000040		
CHR\$M_NUM_BER	=	00000010		
CHR\$M_SPA_MSK	=	00000001		
CHR\$M_SYM_CH1	=	00000008		
CHR\$M_SYM_CHR	=	00000004		
CHR\$M_SYM_DLM	=	00000002		
CHR\$V_COMMA CR	=	00000005		
CHR\$V_CVTLC	=	00000061		
CHR\$V_ILL CHR	=	00000006		
CHR\$V_NOCVT	=	0000007F		
CHR\$V_NUM_BER	=	00000004		
CHR\$V_SPA_MSK	=	00000000		
CHR\$V_SYM_CH1	=	00000003		
CHR\$V_SYM_CHR	=	00000002		
CHR\$V_SYM_DLM	=	00000001		
CLR_XCRF		000003F2	R D	03
CNT	=	00000002		
CR	=	0000000D		
CRODSE		0000039F	RG	03
CROEN/3		00000397	RG	03
CROS)		000003A6	R	03
DSABL		00000286	RG	03
ENABL		00000280	RG	03
ENABL_DSABL		00000288	R	03
ENB\$G_LOCALSYMB	*****		X	03
ENB\$G_LONGNAMES	*****		X	03
ENB\$G_OPTIONS	*****		X	03
ENDPRG		0000045E	RG	03
ERR	=	00000000		
ERRADR		00000058	RG	03

ERRASN	0000002C	RG	03
ERRBLK	00000066	RG	03
ERRBRF	0000007A	RG	03
ERRCHA	0000005F	RG	03
ERRDAR	00000051	RG	03
ERRDOL	00000025	RG	03
ERRENT	00000000	RG	03
ERREXP	0000006F	RG	03
ERRIIF	0000004A	RG	03
ERRMRS	00000015	RG	03
ERRMST	00000033	RG	03
ERROPD	00000007	RG	03
ERRREF	0000003F	RG	03
ERR_0	00000038	R	03
ERR_1	0000003B	R	03
ERR_3	0000006B	R	03
FF	= 0000000C		
FLGSM_ALLCHR	= 00000001		
FLGSM_BOL	= 00000002		
FLGSM_CHKLPND	= 00100000		
FLGSM_COMEXPR	= 00000004		
FLGSM_CONT	= 00000008		
FLGSM_CRF	= 40000000		
FLGSM_CRSEEN	= 00000001		
FLGSM_DATRPT	= 00000010		
FLGSM_DBGOUT	= 00004000		
FLGSM_DLIMSTR	= 00008000		
FLGSM_ENDMCH	= 00000020		
FLGSM_EVALEXPR	= 00000040		
FLGSM_EXPOPT	= 00000080		
FLGSM_EXTERR	= 00010000		
FLGSM_EXTWRN	= 00020000		
FLGSM_FIRSTLN	= 00000200		
FLGSM_IFSTAT	= 00800000		
FLGSM_IIF	= 00400000		
FLGSM_INSERT	= 00000100		
FLGSM_IRPC	= 20000000		
FLGSM_LEXOP	= 00000002		
FLGSM_LSTXST	= 00000200		
FLGSM_MAC2COL	= 00000800		
FLGSM_MACL	= 00000800		
FLGSM_MACLTB	= 08000000		
FLGSM_MACTXT	= 00010000		
FLGSM_MEBLST	= 00001000		
FLGSM_MOREARG	= 00002000		
FLGSM_MOREINP	= 00000008		
FLGSM_NEWPND	= 00000400		
FLGSM_NOREF	= 01000000		
FLGSM_NTYPESPC	= 00000020		
FLGSM_NULCHR	= 00040000		
FLGSM_OBJXST	= 00200000		
FLGSM_OPNDCHK	= 00000100		
FLGSM_OPRND	= 00002000		
FLGSM_OPTVFLIDX	= 00001000		
FLGSM_ORDLST	= 00020000		
FLGSM_P2	= 00004000		
FLGSM_RPTIRP	= 10000000		

MACS
Symbol

AB
AD
AF
AG
AH
AL
AO
AQ
ARG
AUD
AW
B
BLN
CHR
CHR
CHR
CHR
CHR
CHR
CHR
CHR
CHR
CHR
CHR
CHR
CHR
CHR
CHR
CR
D
DANI
DANI
DANI
DAT
DBU
DCL
DCO
DCO
DDI
DEO
DEQ
DFN
DGU
DIN
DIU
DLU
DMA
DMI
DOP
DOP
DOR
DPC
DPL
DPO
DSQ
DSQ

FLGSM_SEQFIL	=	020000000
FLGSM_SKAN	=	00008000
FLGSM_SPECOP	=	00000004
FLGSM_SPLALL	=	04000000
FLGSM_STOIMF	=	00040000
FLGSM_SYM2COL	=	00000400
FLGSM_TOCLFG	=	00080000
FLGSM_UPAFLG	=	00000010
FLGSM_UPDFIL	=	00000080
FLGSM_UPMARG	=	00000040
FLGSM_XCRF	=	80000000
FLGSV_ALLCHR	=	00000000
FLGSV_BOL	=	00000001
FLGSV_CHKLPND	=	00000014
FLGSV_COMPEXP	=	00000002
FLGSV_CONT	=	00000003
FLGSV_CRF	=	0000001E
FLGSV_CRSEEN	=	00000020
FLGSV_DATRPT	=	00000004
FLGSV_DBGOUT	=	0000002E
FLGSV_DLIMSTR	=	0000002F
FLGSV_ENDMCH	=	00000005
FLGSV_EVAEXPR	=	00000006
FLGSV_EXPOPT	=	00000007
FLGSV_EXTERR	=	00000030
FLGSV_EXTWRN	=	00000031
FLGSV_FIRSTLN	=	00000029
FLGSV_IFSTAT	=	00000017
FLGSV_IIF	=	00000016
FLGSV_INSERT	=	00000008
FLGSV_IRPC	=	0000001D
FLGSV_LEXOP	=	00000021
FLGSV_LSTXST	=	00000009
FLGSV_MAC2COL	=	0000002B
FLGSV_MACL	=	0000000B
FLGSV_MACLTB	=	0000001B
FLGSV_MACTXT	=	00000010
FLGSV_MEBLST	=	0000000C
FLGSV_MOREARG	=	00000023
FLGSV_MOREINP	=	00000023
FLGSV_NEWPND	=	0000000A
FLGSV_NOREF	=	00000018
FLGSV_NTYPESPC	=	00000025
FLGSV_NULCHR	=	00000032
FLGSV_OBJXST	=	00000015
FLGSV_OPNDCHK	=	00000028
FLGSV_OPRND	=	0000000D
FLGSV_OPTVFLIDX	=	0000002C
FLGSV_ORDLST	=	00000011
FLGSV_P2	=	0000000E
FLGSV_RPTIRP	=	0000001C
FLGSV_SEQFIL	=	00000019
FLGSV_SKAN	=	0000000F
FLGSV_SPECOP	=	00000022
FLGSV_SPLALL	=	0000001A
FLGSV_STOIMF	=	00000012
FLGSV_SYM2COL	=	0000002A

FLGSV_TOCFLG	=	00000013
FLGSV_UPAFLG	=	00000024
FLGSV_UPDFIL	=	00000027
FLGSV_UPMARG	=	00000026
FLGSV_XCRF	=	0000001F
HASHSZ	=	0000007F
HYPHEN	=	0000002D
IDENT	=	00000092
INPSK_BUFSIZ	=	000003E8
INTSK_BUFSIZ	=	000013F4
INTSK_BUFWRN	=	00001390
INTS_ADD	=	00000001
INTS_AND	=	00000002
INTS_ASH	=	00000003
INTS_ASN	=	0000000C
INTS_AUGPC	=	0000000D
INTS_BDST	=	0000000E
INTS_CHKL	=	0000000F
INTS_DIV	=	00000004
INTS_END	=	00000010
INTS_EPT	=	00000011
INTS_ERR	=	00000012
INTS_ETX	=	00000013
INTS_FNEWL	=	00000014
INTS_ILG	=	00000000
INTS_INFO	=	0000003A
INTS_LGLAB	=	00000015
INTS_MACL	=	00000016
INTS_MUL	=	00000005
INTS_NEG	=	00000006
INTS_NEWL	=	00000017
INTS_NEWP	=	00000018
INTS_NOT	=	00000007
INTS_OP	=	00000019
INTS_OR	=	00000008
INTS_PRIL	=	0000001A
INTS_PRT	=	0000001B
INTS_PSECT	=	0000001C
INTS_REDEF	=	0000001D
INTS_REF	=	0000001E
INTS_REST	=	0000001F
INTS_SAME	=	00000009
INTS_SAVE	=	00000020
INTS_SBTTL	=	00000021
INTS_SETFLAG	=	00000022
INTS_SETLONG	=	00000023
INTS_SPIC	=	00000024
INTS_SPID	=	00000025
INTS_STIB	=	00000026
INTS_STIL	=	00000028
INTS_STIW	=	00000027
INTS_STKEPT	=	00000029
INTS_STKG	=	0000002A
INTS_STKL	=	0000002B
INTS_STKPC	=	0000002C
INTS_STKS	=	0000002D
INTS_STOB	=	00000034

RG 03

[illegible]

MAC\$ACTONE
Symbol table

ONCE-ONLY ACTION ROUTINES

M 1

16-SEP-1984 02:19:08 VAX/VMS Macro V04-00
5-SEP-1984 01:46:56 [MACRO.SRC]ACTONE.MAR;1

Page 19
(11)

```

INT$_STOL      = 0000002E
INT$_STOW      = 00000035
INT$_STRB      = 0000002F
INT$_STRL      = 00000031
INT$_STRSB     = 00000032
INT$_STRSW     = 00000033
INT$_STRW      = 00000030
INT$_STSB      = 00000036
INT$_STSW      = 00000037
INT$_SUB       = 0000000A
INT$_SUME      = 00000039
INT$_WRN       = 00000038
INT$_XOR       = 0000000B
LIST           = 000002FB RG 03
LIST_NLIST     = 00000309 R 03
LST$G_DIRLIST  = ***** X 03
LST$G_LONGNAMES = ***** X 03
LST$K_BUFSIZ   = 00000086
LST$K_L_P_PAGE = 0000003C
LST$K_TITLE_SIZ = 00000028
MAC$AB_HD_NEWPG = ***** X 03
MAC$AB_HD_TITLE = ***** X 03
MAC$AB_HD_TSTRG = ***** X 03
MAC$AB_IDENT   = ***** X 03
MAC$AB_LINEBF  = ***** X 03
MAC$AB_TITLE   = ***** X 03
MAC$AB_TMPSYM  = ***** X 03
MAC$AB_TOC_MSG = ***** X 03
MAC$CLOSE_LIST = ***** X 03
MAC$DEC_OUT_L2X = ***** X 03
MAC$DEC_OUT_R2L = ***** X 03
MAC$DFLT_LIST  = ***** X 03
MAC$DSPL_ARGS  = ***** X 03
MAC$ERRORLN    = ***** X 03
MAC$ERRORPT    = ***** X 03
MAC$ERRORPX    = ***** X 03
MAC$ERR_PUT    = ***** X 03
MAC$GB_MODE    = ***** X 03
MAC$GB_REG     = ***** X 03
MAC$GETCHR     = ***** X 03
MAC$GL_DFPC_DSP = ***** X 03
MAC$GL_DIRF[G] = ***** X 03
MAC$GL_DSLISF  = ***** X 03
MAC$GL_ENLISF  = ***** X 03
MAC$GL_IF_LEVEL = ***** X 03
MAC$GL_INTWRNPT = ***** X 03
MAC$GL_LINEIN  = ***** X 03
MAC$GL_LINEPT  = ***** X 03
MAC$GL_LINE_CNT = ***** X 03
MAC$GL_LIST_IT = ***** X 03
MAC$GL_LIST_LVL = ***** X 03
MAC$GL_LIST_PTR = ***** X 03
MAC$GL_LN_PAGE = ***** X 03
MAC$GL_PC      = ***** X 03
MAC$GL_SRC_PAG = ***** X 03
MAC$GL_TTX_SIZ = ***** X 03
MAC$GL_XFRADR  = ***** X 03

```

```

MAC$GT_SCB      ***** X 03
MAC$GW_LST_INST ***** X 03
MAC$GW_LST_LINE ***** X 03
MAC$INSUSRSYMTB ***** X 03
MAC$INTOUT_2_LW ***** X 03
MAC$INTOUT_N    ***** X 03
MAC$INTOUT_X    ***** X 03
MAC$K_HD_SIZE   ***** X 03
MAC$LIST_RAB    ***** X 03
MAC$OUTFRAME    ***** X 03
MAC$PASS1_END   ***** X 03
MAC$SET_NEW_LSB ***** X 03
MAC$SET_PC      ***** X 03
MAC$SKIPSP      ***** X 03
MAC$SKP_OPR     ***** X 03
MAC$SRCOSRSYMTB ***** X 03
MAC$SRC_LIST    ***** X 03
MAC$SYMSCNUP    ***** X 03
MAC$_ADRLSTS_YN = 007D9002
MAC$_ASGNMNTSYN = 007D9022
MAC$_BADENTRY   = 007D902A
MAC$_BLKDIRSYN  = 007D9042
MAC$_DATA_LSYN  = 007D9062
MAC$_DIRSYN     = 007D906A
MAC$_ILLASCARG  = 007D90B2
MAC$_ILLDFLTARG = 007D90CA
MAC$_ILLEXPR    = 007D90D2
MAC$_ILLOPDEF   = 007D910A
MAC$_ILLSYMLEN  = 007D8820
MAC$_MCHINSTSYN = 007D914A
MAC$_MSGCMAIIF  = 007D9152
MAC$_NOTENABOPT = 007D9172
MAC$_NOTINMACRO = 007D918A
MAC$_NOTLGLISOP = 007D9192
MAC$_OPRNDSYN   = 007D91A2
MAC$_REGOPSYN   = 007D91CA
MAC$_UNDEFXFRAD = 007D921A
MAC$_UNRECSTMT  = 007D9222
MAC$_UNTERMARG  = 007D922A
MAC$_UNTERMCOND = 007D9232
MAC_SUBSYS      = 0000007D
NLIST           = 00000303 RG 03
OBJ$K_BUFSIZ    = 00000200
OPF$M_LASTOPR   = 00002000
OPF$M_OPTEXP    = 00001000
OPF$V_LASTOPR   = 0000000D
OPF$V_OPTEXP    = 0000000C
PSC$B_NAME      = 00000004
PSC$B_SEG       = 0000000C
PSC$B_UNUSED    = 0000000B
PSC$K_BLK_SIZ   = 00000013
PSC$K_NO_OPTS   = 0000000A
PSC$L_CURLOC    = 0000000F
PSC$L_LINK      = 00000000
PSC$L_MAXLGTH   = 00000005
PSC$M_ABS       = FFFFFFFF7
PSC$M_ALIGNFLG  = 00004000

```

MAC\$
Symt

SYM\$
SYM\$
SYM\$
SYM\$
TAB
VB
VD
VF
VG
VH
VL
VO
VQ
VW
W
WB
WD
WF
WG
WH
WL
WO
WQ
WW
X1
X2

PSEC

.
.
.
\$AB\$
MAC\$
MAC\$

Pha\$

Init
Com\$
Pas\$
Sym\$
Pas\$
Sym\$
Pse\$
Cro\$
Ass\$
The
276\$

MAC\$ACTONE
Symbol table

ONCE-ONLY ACTION ROUTINES

N 1

16-SEP-1984 02:19:08 VAX/VMS Macro V04-00
5-SEP-1984 01:46:56 [MACRO.SRC]ACTONE.MAR;1

Page 20
(11)

PSC\$M_ALLOPTNS = 000003FF
PSC\$M_BYTE = 00004000
PSC\$M_CON = FFFFFFFFB
PSC\$M_DEFAULT = 000001C8
PSC\$M_EXE = 000000C0
PSC\$M_GBL = 00000010
PSC\$M_LCL = FFFFFFFEF
PSC\$M_LIB = 00000002
PSC\$M_LONG = 00004800
PSC\$M_NOEXE = FFFFFFFBF
PSC\$M_NOPIC = FFFFFFFFE
PSC\$M_NORD = FFFFFFF7F
PSC\$M_NOSHR = FFFFFFFDF
PSC\$M_NOVEC = FFFFFFFDF
PSC\$M_NOWRT = FFFFFFFEF
PSC\$M_OVR = 00000004
PSC\$M_PAGE = 00006400
PSC\$M_PIC = 00000001
PSC\$M_QUAD = 00004C00
PSC\$M_RD = 00000080
PSC\$M_REL = 00000008
PSC\$M_SHR = 00000020
PSC\$M_USR = FFFFFFFFD
PSC\$M_VEC = 00000200
PSC\$M_WORD = 00004400
PSC\$M_WRT = 00000180
PSC\$S_ALIGNMENT = 00000004
PSC\$V_ALIGNFLG = 0000000E
PSC\$V_ALIGNMENT = 0000000A
PSC\$V_EXE = 00000006
PSC\$V_GBL = 00000004
PSC\$V_LIB = 00000001
PSC\$V_OVR = 00000002
PSC\$V_PIC = 00000000
PSC\$V_RD = 00000007
PSC\$V_REL = 00000003
PSC\$V_SHR = 00000005
PSC\$V_VEC = 00000009
PSC\$V_WRT = 00000008
PSC\$W_FLAG = 00000009
PSC\$W_OPTIONS = 0000000D
RAB\$R_RBF = 00000028
RAB\$W_RSZ = 00000022
RDX\$V_BINARY = 00000000
RDX\$V_DECIMAL = 00000002
RDX\$V_DOUBLE = 00000005
RDX\$V_FLOAT = 00000004
RDX\$V_GFLOAT = 00000006
RDX\$V_HEX = 00000003
RDX\$V_HFLOAT = 00000007
RDX\$V_OCTAL = 00000001
REG\$PC = 0000000F
SBTTC = 00000156 RG 03
SBT_PUT_LIN = 00000268 R 03
SEMT = 0000003B
SETDFL = 000003F8 RG 03
SET_XCRF = 000003EC R D 03

SIZ... = 00000001
STB\$K_PG_MISS = 0000000A
SUM_B_FLAGS = 0000001C
SUM_K_BLN = 0000001D
SUM_L_ISDATA = 00000004
SUM_L_STS = 00000000
SUM_M_AUDIT = 00000001
SUM_M_AUDITNEW = 00000002
SUM_M_DELETE = 00000010
SUM_M_SRCUPD = 00000004
SUM_M_SUBCLSH = 00000008
SUM_Q_AUDDS = 00000008
SUM_Q_FILESP = 00000010
SUM_V_AUDIT = 00000000
SUM_V_AUDITNEW = 00000001
SUM_V_DELETE = 00000004
SUM_V_SRCUPD = 00000002
SUM_V_SUBCLSH = 00000003
SUM_W_INSERT_NO = 0000001A
SUM_W_LINE_NO = 00000018
SYMSB_NAME = 00000004
SYMSB_SEG = 0000000C
SYMSB_TOKEN = 0000000B
SYMSK_BLKSIZE = 0000000D
SYMSK_MAXLEN = 0000001F
SYMSK_TWOCOL = 00000010
SYMSL_LINK = 00000000
SYMSL_VAL = 00000005
SYMSM_ABS = 00000010
SYMSM_ASN = 00000100
SYMSM_CRFO = 00002000
SYMSM_DEBUG = 00000020
SYMSM_DEF = 00000001
SYMSM_DELMAC = 00000200
SYMSM_EPT = 00000200
SYMSM_EXTRN = 00000008
SYMSM_GLOBL = 00000004
SYMSM_LOCAL = 00000040
SYMSM_ODBG = 00000400
SYMSM_REF = 00000080
SYMSM_RELPSECT = 00000800
SYMSM_SUPR = 00004000
SYMSM_WEAK = 00000002
SYMSM_XCRF = 00001000
SYMSV_ABS = 00000004
SYMSV_ASN = 00000008
SYMSV_CRFO = 0000000D
SYMSV_DEBUG = 00000005
SYMSV_DEF = 00000000
SYMSV_DELMAC = 00000009
SYMSV_EPT = 00000009
SYMSV_EXTRN = 00000003
SYMSV_GLOBL = 00000002
SYMSV_LOCAL = 00000006
SYMSV_ODBG = 0000000A
SYMSV_REF = 00000007
SYMSV_RELPSECT = 0000000B

MAC\$
VAX-

Ther
231
14 p

Macr

\$25
- \$25
TOTAL

537
Ther
MACF

MAC\$ACTONE
Symbol table

ONCE-ONLY ACTION ROUTINES

B 2

16-SEP-1984 02:19:08 VAX/VMS Macro V04-00
5-SEP-1984 01:46:56 [MACRO.SRC]ACTONE.MAR;1

Page 21
(11)

SYMSV_SUPR = 0000000E
SYMSV_WEAK = 00000001
SYMSV_XCRF = 0000000C
SYMSW_FLAG 00000009
SYSPOT ***** GX 03
TAB = 00000009
TITLE 000000EE RG 03
X = 00000010
X1 = 00000400
X2 = 0000000F

+-----+
! Psect synopsis !
+-----+

PSECT name	Allocation	PSECT No.	Attributes
. ABS .	00000000 (0.)	00 (0.)	NOPIC USR CON ABS LCL NOSHR NOEXE NORD NOWRT NOVEC BYTE
. BLANK .	00000000 (0.)	01 (1.)	NOPIC USR CON REL LCL NOSHR EXE RD WRT NOVEC BYTE
\$ABSS	0000001D (29.)	02 (2.)	NOPIC USR CON ABS LCL NOSHR EXE RD WRT NOVEC BYTE
MAC\$RO_CODE_P1	00000499 (1177.)	03 (3.)	NOPIC USR CON REL GBL NOSHR EXE RD NOWRT NOVEC LONG

+-----+
! Performance indicators !
+-----+

Phase	Page faults	CPU Time	Elapsed Time
Initialization	29	00:00:00.02	00:00:02.32
Command processing	121	00:00:00.44	00:00:03.08
Pass 1	270	00:00:04.92	00:00:22.15
Symbol table sort	0	00:00:00.61	00:00:02.33
Pass 2	132	00:00:01.32	00:00:05.61
Symbol table output	55	00:00:00.24	00:00:01.52
Psect synopsis output	1	00:00:00.02	00:00:00.02
Cross-reference output	0	00:00:00.00	00:00:00.00
Assembler run totals	610	00:00:07.58	00:00:37.04

The working set limit was 1500 pages.
45183 bytes (89 pages) of virtual memory were used to buffer the intermediate code.
There were 40 pages of symbol table space allocated to hold 649 non-local and 50 local symbols.
630 source lines were read in Pass 1, producing 24 object records in Pass 2.
24 pages of virtual memory were used to define 22 macros.

+-----+
! Macro library statistics !
+-----+

Macro library name	Macros defined
-\$255\$DUA28:[SHRLIB]SUM.MLB;1	3
-\$255\$DUA28:[MACRO.OBJ]MACRO.MLB;1	12
-\$255\$DUA28:[SYSLIB]STARLET.MLB;2	7
TOTALS (all libraries)	22

754 GETS were required to define 22 macros.

MAC\$ACTONE
VAX-11 Macro Run Statistics

ONCE-ONLY ACTION ROUTINES

C 2

16-SEP-1984 02:19:08 VAX/VMS Macro V04-00
5-SEP-1984 01:46:56 [MACRO.SRC]ACTONE.MAR;1

Page 22
(11)

There were no errors, warnings or information messages.

MACRO/LIS=LIS\$:ACTONE/OBJ=OBJ\$:ACTONE MSRC\$:ACTONE/UPDATE=(ENH\$:ACTONE)+LIB\$:MACRO/LIB+SHRLIB\$:SUM/LIB

0223 AH-BT13A-SE
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY

0224

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY